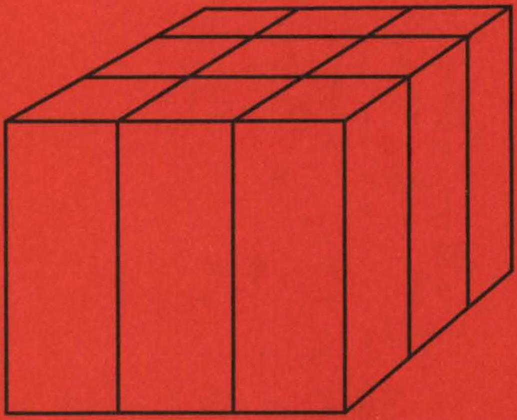
IBM

VSE/Advanced Functions Diagnosis Reference

Linkage Editor



VSE/Advanced Functions Diagnosis Reference

Linkage Editor

Program Number 5666-301

Order Number LY33-9112-0
File No. S370/4300-31

First Edition (March 1985)

This edition applies to Version 2 Release 1 of VSE/Advanced Functions, Program Number 5666-301 and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters. Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest IBM System/370, 30XX and 4300 Processors Bibliography, GC20-0001, for the editions that are applicable and current.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM program product in this document is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

Publications are not stocked at the addresses given below; requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed either to:

IBM Corporation
Dept. 6R1
180 Kost Road
Mechanicsburg, PA 17055, USA

or to:

IBM Deutschland GmbH
Dept. 3248
Schoenaicher Strasse 220
D-7030 Boeblingen, Federal Republic of Germany

IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

PREFACE

This Diagnosis Reference documents the code of the linkage editor component of VSE/Advanced Functions for the task of program service.

The manual consists of the following chapters:

- INTRODUCTION to the linkage editor, showing the context of the component in the system, the input and output of the program, and the different ways of calling it.
- DESIGN INFORMATION, describing the function, I/O flow, storage layout, and division into CSECTs and modules of the program. A description of the linkage editor phase gives all relevant information from the prologue of the listing. A detailed description shows the sequence of operations for each CSECT with the essential labels.
- ORGANIZATION INFORMATION, listing the flow of control between the CSECTs of the program.
- DATA AREAS, describing the formats of library records used by the program, linkage editor tables, and the librarian areas used by the linkage editor.
- DIAGNOSTICS which contains interface information and cross references.

An index at the end of the manual will help to find details fast.

The text refers to the following IBM manuals:

- VSE/Advanced Functions System Control Statements, SC33-6198
- VSE/Advanced Functions Service Aids, SC33-6195
- VSE/Advanced Functions Diagnosis Reference: Librarian, LY33-9111

Titles and abstracts of other related publications are listed in System/370, 30xx and 4300 Processors Bibliography, GC20-0001.

SUMMARY OF AMENDMENTS

Version 2 Release 1 has the following technical improvements that are documented in this manual:

- Changes due to the new librarian concept in VSE/Advanced Functions.
- Virtual I/O for temporary storage of phases between link-editing and following execution.

The material has been reorganized in the form of Diagnosis Reference manuals.

CONTENTS

FIGURES

INTRODUCTION

System Context

All programs to be executed under VSE/Advanced Functions must be prepared first by a language translator and then by the linkage editor. According to these stages of preparation, programs may be stored in any sublibrary as one of the predefined member types.

> Source modules have a one-character type (A-Z,0-9,#,$,@).
>   (Each compiler defines its valid source module types.)
> Object modules have type OBJ.
> Phases have type PHASE.

The linkage editor can run in any partition.

Input

Input consists of the linkage editor statements and object modules to be linked. Linkage editor statements are ACTION, ENTRY, INCLUDE, and PHASE. For the specification of these statements see VSE/Advanced Functions System Control Statements, SC33-6095. The sequence of sublibraries to be searched for the input object modules is defined by a LIBDEF OBJ,SEARCH job statement. This sequence is the OBJ-search chain and a sublibrary in this chain is called a SEARCH sublibrary.

Output

Output consists of the phase or phases produced and a linkage editor map giving address information about each phase and CSECT. The phase produced is stored either temporarily or permanently before it can be executed. For // OPTION LINK the phase is stored temporarily in the virtual I/O area, an extension of the page data set. For // OPTION CATAL the phase is stored permanently in the CATALOG sublibrary with an entry in the sublibrary member index. The CATALOG sublibrary is defined via the LIBDEF PHASE,CATALOG job statement. The member index is updated via library management. The linkage editor map is printed on SYSLST after the linkage editor control statements and input listing.

Invocation

Job control calls the linkage editor when it reads a // EXEC LNKEDT statement. This statement can occur in two different combinations:

1. To link and catalog: An object module is link-edited and the resulting phase is permanently stored or "cataloged" in a sublibrary. If the phase is re-enterable it can be declared SVA-eligible in the PHASE statement. The job statements to "link and catalog" are:

   ```
   // OPTION CATAL
   // EXEC LNKEDT
   ```

2. To link, load, and execute: An object module is link-edited and the resulting phase is temporarily stored in the virtual I/O area and immediately executed. The phase to be link-edited and immediately executed must not be part of an overlay structure. The job statements to link, load, and execute are:

   ```
   // OPTION LINK
   // EXEC LNKEDT
   // EXEC
   ```

MSHP Module Control

Phases cataloged under control of MSHP must be maintained via MSHP. Such phases are flagged in the directory entry as being MSHP controlled. This flag normally prevents the replacement of the phase by the linkage editor when not running under control of MSHP.
To allow replacement of such a phase out of control of MSHP, the MSHP bypass function is provided. If this bypass is specified, the linkage editor replaces MSHP-flagged phases and sets another flag to indicate that the phase has been changed without MSHP control.

The MSHP bypass is specified as PARM parameter as follows:

```
// EXEC LNKEDT,PARM='MSHP'
```

Controlled Operator Cancel

Cancelling a librarian or linkage editor job in a prior release was always critical since the library used was in danger to be destroyed. For Version 2 Release 1 of VSE/Advanced Functions the librarian or linkage editor continues processing up to a point where a consistent state of the library is reached whenever a librarian command or a linkage editor job is cancelled normally, i.e. not with CANCEL FORCE.
CANCEL FORCE always terminates immediately with risk of library damage.

The normal cancel function is controlled by the flags IJBARCNA and IJBCNCPD in field JCSW8 of the partition COMREG. These flags are set/reset/checked by the linkage editor module INLPLEIT.

DESIGN INFORMATION

FUNCTION

The linkage editor adjusts the addresses in the CSECTs of one or several object modules so that these CSECTs can be executed together as one phase or as several phases in an overlay structure.

Input

Input to the linkage editor are the linkage editor statements and the object modules produced by language translators (assembler or compiler).

Each object module consists of dictionaries (ESD and RLD) and text for one or more control sections.

Six record types can be produced by the language translators out of the source input to form a module. They appear in the following order:

Rec. Type	Definition
ESD	External symbol dictionary
SYM	Ignored by the linkage editor
TXT	Text
RLD	Relocation list dictionary
REP	Replacement to text by programmer
END	End of module

The dictionaries contain the information necessary for the linkage editor to resolve references between different modules. The text consists of executable instructions and data fields.

Figure 1 shows an overview of the linkage editor input.

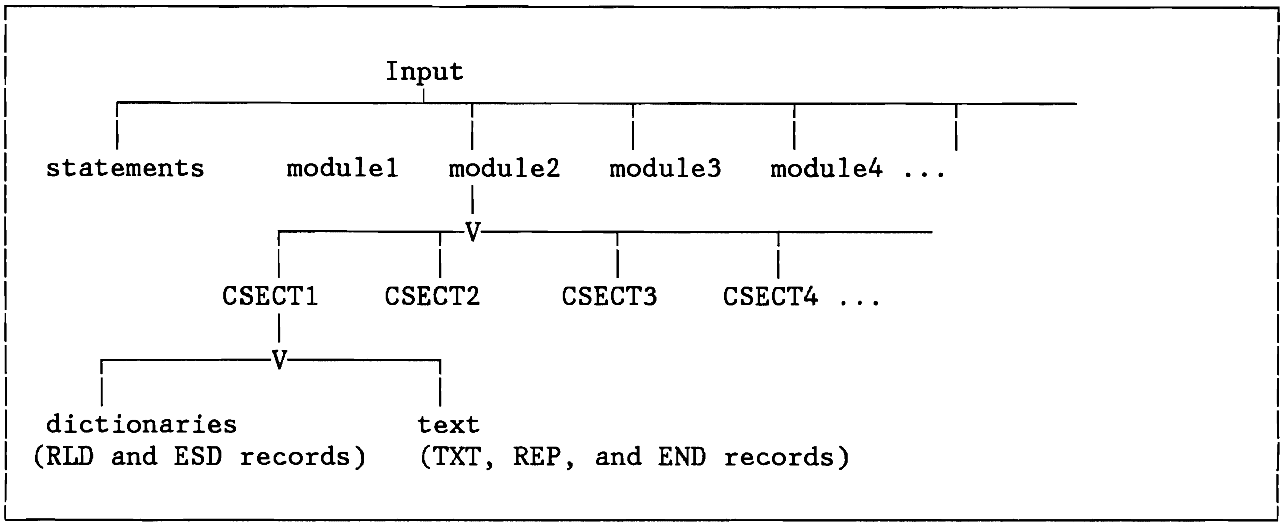


Figure 1. Overview of Input Units

The linkage editor can also re-link phases if they have been retransformed into modules by the librarian command PUNCH. In this way, an already link-edited phase can be recataloged to another library and sublibrary. The librarian command PUNCH causes the contents of a member of type PHASE to be punched as TXT and RLD records.

Since the greatest task of the linkage editor is the adjustment of external symbols, ESD records have the most variety.

ESD RECORDS: The following types of ESD records exist.

- Section Definition (SD): Consists of CSECT name, assembled origin, and length of a named CSECT. Generated by a START or CSECT statement in the source module.

- Private Code (PC): Consists of assembled origin and length of an unnamed CSECT.

- Label Definition or Label Reference (LD/LR): Contains the assembled address and the associated SD of a label that may be referred to by another module. The LD entry is termed LR (label reference) when the entry is matched to an ER entry.

- External Reference (ER): Contains the location of a reference made to another module. Generated by the assembler instruction EXTRN or a V-type address constant in the source module.

- Weak External Reference (WX): Same content as ER, except that WX suppresses Autolink. Generated by the assembler instruction WXTRN.

- Common (CM): Contains the number of bytes of "common storage" needed by a particular phase at execution time. Common storage is an area to be reserved at the beginning of the partition for shared use between phases. Its most frequent use is for communication in an overlay structure. Generated in the source module by the assembler instruction COM or directly from the compilers.

Output

The output consists of the phases produced and a linkage editor map giving address information about each phase. See the description of the linkage editor map in the Chapter "Diagnostics".

Operation

The program takes the CSECTs out of several modules and combines them, in a different selection or sequence if so specified, into executable phases.

Figure 2 shows how phases can be formed.

Sample of a two-module input resulting in a three-phase output

Language Translator Output	Linkage Editor Output
Module A	Phase 1
ESDs	CSECTA
TXT-CSECTA	CSECTB
TXT-CSECTB	
TXT-CSECTC	Phase 2
RLDs	CSECTC
End	CSECTD
	CSECTE
Module B	
ESDs	Phase 3
TXT-CSECTD	CSECTA
TXT-CSECTE	CSECTF
TXT-CSECTF	CSECTG
TXT-CSECTG	CSECTD
RLDs	
END	

Figure 2. Example of a Module-Phase Relationship

When the linkage editor reads a module, it stores the ESD records in its control dictionary (CD), writes the information from the TXT and

REP records into the sublibrary, and the RLD items on an internal buffer called RLD buffer. If the RLD buffer is full, its content is written on workfile IJSYS01.

The CD contains the information to find each CSECT and to resolve any reference between CSECTs.

The language translator gives each ESD record a number called ESID number. The linkage editor gives it a CD number unique in the phase, because the same ESID number might occur several times coming from the different modules.

In detail, the linkage editor does the following:

1. Relocates the origin of each CSECT in the phase:
   The relocation factor for each CSECT is determined and saved in the CD.

2. Assigns to each phase an area of storage, determines its transfer address, and combines the module TXT records into phase blocks using librarian services.

3. Resolves address cross references across CSECTs or phases in overlay structures.

4. Adjusts the contents of the address constants (ADCONs) in the phase and inserts them in the text. These ADCONs are identified in the RLD. Unresolved ADCONs appear in the phase as zero RLD items.

As we have seen above, the linkage editor allows the inclusion of the same control section (CSECT) within each of several phases. If a CSECT appears in a ROOT phase, it does not appear in any other phase. (This does not apply to CSECTs that begin with the letters IBM.) A duplicate CSECT within the same phase will be ignored.

The following examples show how external references are resolved, depending on whether or not a ROOT phase exists. The first example shows how external references are resolved when there is no ROOT phase:

```
PHASE ONE
   CSECT A
    V(B)──────┐
              V
   CSECT B<───┴────┐
                   │
PHASE TWO          │
   CSECT A         │
    V(B)───────────┘
   CSECT B<───┐
              │
PHASE THREE   │
   CSECT A    │
    V(B)──────┘
   CSECT B

PHASE FOUR
   CSECT A
   CSECT B<──┐
             │
    V(B)─────┘
```

The second example shows the resolution of external references with a ROOT phase:

```
ROOT-PHASE ZERO
   CSECT V
    ENTRYPOINT X<──┬────┐
                   A    │
PHASE ONE          │    │
   CSECT A         │    │
    V(X)───────────┘    │
   CSECT B              │
    ENTRYPOINT X        │
                        │
PHASE TWO               │
   CSECT A              │
   CSECT B              │
    ENTRYPOINT X        │
    V(X)────────────────┘
```

Privileged external references (names beginning with the letters IJ or IBM) are always resolved within the current phase or the ROOT phase. If this is not possible, the resolution will be attempted at the end of the phase via the AUTOLINK function (if NOAUTO is specified, the IJ or IBM prefix is not privileged). The other previously defined phases are not examined for possible resolution. If an external reference does not match the name of a module in the sublibraries to be searched, it will be an unresolved external reference.

The following example shows the resolution of privileged external reference:

```
   PHASE ONE
      CSECT A
       V(IJxx)─────────┐
       V(IBMx)──────┐  │
       V(B)──────┐  │  │
                 │  │  │
┌────>CSECT B<───┘  │  │
│     CSECT IBMx<───┘  │
│     CSECT IJxx<──────┘
│
│  PHASE TWO
│     CSECT A
│      V(IJxx)─────────┐
│      V(IBMx)──────┐  │
└──────V(B)         │  │
      CSECT B       │  │
      CSECT IBMx<───┘  │
      CSECT IJxx<──────┘
```

Autolink

Autolink is a feature that works automatically when a new PHASE or an ENTRY statement indicates that the preceding phase is finished, unless the user has suppressed it by a PHASE NOAUTO statement, by an ACTION statement option, or by the assembler instruction WXTRN in the source module. It is located in the CSECT IJBLNK together with the general input control.

What it Does

Autolink tries to find in the sublibrary OBJ-search chain a module to resolve any unresolved-external reference (ER) from the preceding phase.

How it Operates

Whenever an external symbol (ER) is encountered Autolink searches the sublibraries specified in the OBJ-search chain for the module of the name indicated by the ER. The module is then treated as an INCLUDE statement, that means, it must also contain an entry with the same name as the ER which is being resolved.

How the Linkage Editor Calculates Addresses

The assembled origin is the address of a CSECT within its object module. This assembled CSECT origin can be declared by assembler statements. Default is 0 for the first, 0 + length of the first for the second CSECT, and so on.

```
0           500              1500
┌───────────┬───────────────┬──────────
│  CSECT 1  │       CSECT 2 │  CSECT 3
└───────────┴───────────────┴───────────
```

The phase origin is the address to which the phase is linked. This address can be specified in the PHASE statement. The default is the beginning address of the linkage editor partition plus the length of the save area.

The expression "phase origin" is used also for the beginning of each CSECT after it was linked into a phase.

The difference between the assembled origin and the phase origin then is called the relocation factor.

```
┌──Start of Virtual Storage
│             ┌─────Start of Linkage Editor Partition
│             │         ┌────────────────────Phase Origin
│             │         │
V─────────────V─────────V───────────────
│             │ save area │ phase------->
├───────────────────────┼──────────────
|<----relocation factor---->| (if assembled origin was 0)
```

If the assembled origin of the phase was 0, the relocation factor is the amount of the address of the phase origin. If it was 500, the relocation factor is the phase origin minus 500.

If the phase is relocatable and is loaded somewhere else later, the address where it is loaded is called the load origin. And now the relocation factor is the new difference between the assembled origin and the load origin.

In general, the following formula is true:

assembled origin + relocation factor = phase or load origin

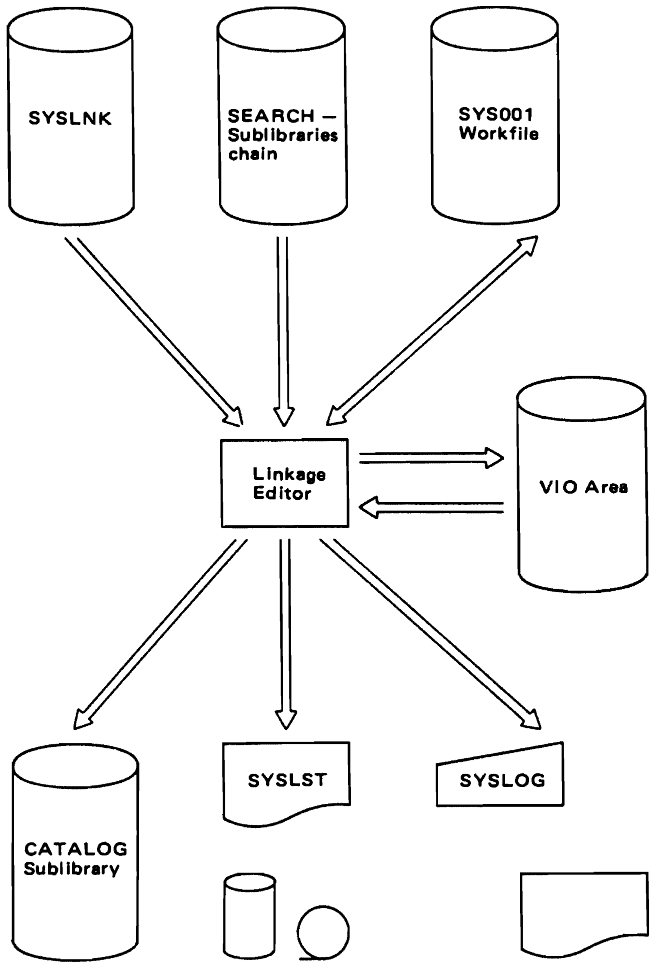
I/O FLOW AND LIBRARY ACCESS CONTROL

As explained in the introduction to this manual, library access is gained for the linkage editor by job control via the LIBDEF statement. Input modules are accessed by a LIBDEF OBJ,SEARCH statement. The CATALOG sublibrary to which the output phases are written is defined via LIBDEF PHASE,CATALOG statement.

The CSECTs IJBINL, IJBOTH, IJBCAT, IJBRLD, and IJBFIN, each call the interface CSECT INLPLEIT which handles all the interfaces with the librarian via librarian macros and control blocks, the stow table, and service buffers for reading from and writing to the sublibraries and the CATALOG sublibrary member index.

The temporary output of a phase to the VIO area is done by the virtual I/O (VIO) routines of the supervisor which are called also by CSECT INLPLEIT using the macro VIO with the operands OPEN, MOVE, or EXTND.

Figure 3 shows the I/O flow in the linkage editor program.



Input comes from:	Output goes to:
SYSLNK - Control stmts and modules	CATALOG sublibrary
OBJ-search chain of sublibraries for input modules	VIO area for phases to be executed immediately
IJSYS01 - Workfile for RLD records	SYSLST - List device
	SYSLOG - System console

Figure 3. I/O Flow

PROGRAM LAYOUT IN THE PARTITION

As Figure 4 shows, the program in the partition starts out with the root of the phase, CSECT IJBLNK. Then follow all other CSECTs of which CSECT (and module) INLPLEIT is the last one. It is followed by work buffers and tables, and the CD with a fixed beginning but variable length. The lengths of the P- and S-buffers are calculated from the high end of the partition down to the end of the CD.

	Program Part:		Size:
Partition Start	IJBLNK and all other CSECTs		
	SYSLNK Buffer LIFO Stack RLD Buffer		328 bytes 210 bytes 2K bytes
	Linkage Table	Stow Table	1540 bytes
	Control Dictionary		variable
	P-Buffer for reading object modules		9496 or 17532
	P-Buffer for handling phases	SMAP processing	See following figure
Partition End	S-Buffer for librarian access		4904 or 6052

Areas	Pointer Names in CSECT IJBLNK
SYSLNK buffer	FLNBUF
LIFO stack	LCSTBEG
RLD buffer for IJSYS01	RLDBUF
Linkage tables	LTMINE
Control dictionary (CD)	CDENT1
Last entry in CD	CTLDAD
Upper limit for CD	TENK
P-buffer for phases	PHAPBUF
P-buffer for modules	OBJPBUF
S-buffer	SHARBUF

Figure 4. Partition Layout

The S-buffer and the P-buffers are formatted by the librarian. The S-buffer is the shared buffer for library management of the CATALOG sublibrary; the P-buffers are used for member I/O.
(For a complete description of the S- and P-buffers see VSE/Advanced Functions Diagnosis Reference: Librarian LY33-9111)

Figure 5 shows how the buffer sizes depend on the partition size.

Partition	S-Buffer	P-Buff.for Mod.	P-Buffer for Phase
up to 256K	4904	9496	3/4 of storage between start of CD and start of S-Buffer, minus length of P-Buffer for modules (up to a maximum of 150K)
over 256K	6052	17532	

Figure 5. Approximate Buffer Sizes in Relation to Partition Size

FUNCTION-TO-CSECT OVERVIEW

The program consists of one phase $LNKEDT. This phase consists of two modules, INLPLEIT and IJBLE1. INLPLEIT consists of one CSECT of the same name; IJBLE1 consists of a number of CSECTS whose names all begin with IJB. The functions of the CSECTs, in the order as they are called, are the following:

IJBINL Serves as entry point to the program. Initializes linkage editor processing. Processes ACTION statements.

INLPLEIT Calls librarian routines or VIO routines of the supervisor.

IJBLNK Contains common subroutines and constants. Processes linkage editor input. Branches to appropriate CSECT. Performs Autolink.

IJBOTH Processes TXT, REP, RLD, and END records.

IJBFIN Finds records and modules from SYSLNK and from SEARCH sublibraries. Formats messages.

IJBESD Processes ESD records.

IJBSCN Processes INCLUDE, PHASE, and ENTRY statements.

IJBCTL Pre- and post-processes PHASE and ENTRY statements.

After ENTRY-statement:

IJBMAP Prints linkage editor map.

IJBRLD Relocates ADCONs in each phase and generates the corresponding RLD records to be used by FETCH/LOAD or by the linkage editor if the phase is re-linked.

IJBCAT Updates the member index of the sublibrary.

PHASE DESCRIPTION FOR PHASE $LNKEDT

ENTRY POINT: IJBINL

FUNCTION: Transforms modules of a program into executable phases.

(For details see "Sequence of Operation" in the CSECTs.)

CALLED BY: Job Control or MSHP

PHASES CALLED: $IJBLBR (Librarian)

DATA AREAS USED:

- I/O areas
- Linkage tables
- Control dictionary
- LEITPL (Communication area to librarian)
- Buffers for librarian services
- Stow table (librarian)
- LIFO stack
- Librarian interface control blocks used by INLPLEIT

MESSAGES CAUSED: See Message-to-CSECT Cross Reference

MESSAGES ISSUED: All by CSECT IJBLNK

INPUT: Statements and object modules

OUTPUT: Phases and linkage editor map

EXIT NORMAL: Caller

EXIT ERROR: Cancel

REGISTER USE:

R10 Base Register for IJBLNK
R11 Base Register for IJBFIN
R10/11/12 Base Registers for INLPLEIT
R13 Base register for all other CSECTs

SEQUENCE OF OPERATION: The sequence of operation is listed in the following by CSECT, in the order of their execution. The most important labels are given on the left side and the operation which they mark on the right side.

CSECT IJBINL

Called By: Job Control

```
  IJBINL  : - Calculates relocation factor.
  OPCATAL : - Opens SYSLST and SYSLNK.
  CALCWA  : - Calculates storage addresses for:
              -- I/O areas
              -- Linkage table
              -- Control dictionary
              -- Buffers for librarian services
              -- Stow table
            - Gets partition start address and checks size.
  CONCLIB : - Calls INLPLEIT to get access to CATALOG sublibrary.
  INLN000 : - Reads from SYSLNK.
  NXTCARD : - Processes ACTION statement if there is one.
  INTER100: - On finding a non-ACTION statement ----> IJBLNK
```

CSECT INLPLEIT

Called By: IJBINL, OTH, ESD, FIN, RLD, CAT

CSECT INLPLEIT uses or modifies the following librarian and supervisor macros:

INLMFIND
INLMGETR
INLMLAMB
INLMLDIS
INLMLRPL
INLMMCON
INLMMDIS
INLMNOTE
INLMPOIN
INLMPUTR
INLMSCON
INLMSTOW
LBRACCCB
LBRACCES

VIO EXTND
VIO MOVE
VIO OPEN

and the following librarian and supervisor control blocks:
INLCDENT
INLCLAMB
INLCLARG
INLCLPT (field LPTVIOTB)
INLCLRPL
INLCMACB
INLCSACB
LBRACCDS

MAPVIORB

This is how it operates:

OPENC : - Builds CATALOG sublibrary control blocks using the macros:
INLCDENT to generate structure declarations for the control blocks INLCDENT (stow table entry)
INLCLARG (any search argum.)
LBRACCCB to initialize the control block LBRACCDS
INLMLAMB to initialize the control block INLCLAMB
INLMLRPL to initialize the control block INLCLRPL
LBRACCES to connect to the specified CATALOG sublib.
INLMSCON to initialize the control block INLCSACB
- Sets flag IJBARCNA in partition COMREG to establish the delayed cancel function in order to keep a consolidated library structure for a normal cancel request.
- Opens virtual I/O control block if option was LINK using the macro VIO OPEN which initializes the control block MAPVIORB. The MAPVIORB pointer is inserted into the table pointed to by field LPTVIOTB in control block INLCLPT. This table is in the system GETVIS area.

OPENJ : - Builds library control block INLCLRPL using the macro INLMLRPL for requests to the OBJ-search chain.

CONNECT: - Uses routine LEITFIND to establish if a phase of the same name as the output phase is in some sublibrary.
If MSHP is active, the phase is replaced in the same sublibrary.
If MSHP is not active and the phase is not under MSHP control, the phase is replaced in the specified CATALOG sublibrary.
- Uses the macro INLMMCON to build control block INLCMACB.

WRITE : - Writes text information into CATALOG sublibrary using macro INLMPUTR.
- Writes text information into VIO area, if option LINK, using macro VIO MOVE. IF necessary, extends the virtual storage allocation using macro VIO EXTND.

UPDATE : - Updates ADCONs in already processed phases using macros INLMGETR and INLMPUTR.
- Updates ADCONs in phase already in VIO area, if option LINK, using macro VIO MOVE.

READ : - Reads OBJ records from object modules using macro INLMGETR.

FINDM : - Finds member from OBJ-search chain using macro INLMFIND.

NOTEM : - Notes position in object module using macro INLMNOTE.

POINTM : - Re-positions in OBJ member using macro INLMPOIN.

STOWML : - Catalogs phases specified in the stow table using macro INLMSTOW.

LEITFIND: - Establishes target sublibrary by finding the phase to be replaced, if the linkage editor runs under MSHP.
- If phase is MSHP controlled, allows replace only under MSHP.
The routine uses the macros INLMFIND, INLMMDIS, INLMLDIS, LBRACCES, and INLMSCON.

CSECT IJBLNK

Called By: IJBINL, SCN, CTL, OTH, ESD, FIN

```
CDENT1  : - Points to control dictionary (CD) in partition
LTMINE  : - Points to Linkage Table in partition
SHARBUF : - Points to buffer for CATALOG sublibrary access
PHAPBUF : - Points to buffer for CATALOG sublibrary access
OBJBUF  : - Points to buffer for OBJ-search chain access
LEITPL  : - Communication area between INLPLEIT and other CSECTs
LCSTBEG : - Header of LIFO stack for nested INCLUDEs
CPHENT  : - Current phase entry
CESDENT : - Current ESD entry

ALNKPR  : - Searches the control table for unresolved ER's and
            initializes for Autolink.
RDNEXT  : - Reads "cards" from SYSLNK or SEARCH sublibrary.
RDEXEC  : - Passes control to:
            ---------------> IJBSCN (Control statement)
            ---------------> IJBESD (ESD records)
            ---------------> IJBOTH (TXT,REP,RLD records)
```

Subroutines in IJBLNK

As the first CSECT (IJBLNK) of the linkage editor program, IJBLNK contains most of the subroutines used by other linkage editor CSECTs. After processing any of these subroutines, control is returned to the calling routine if not indicated otherwise.

The following list shows name, main entry points, and function of each routine.

```
Subr.   Entry   Function
LTESID: Input to this routine is an ESID number supplied by the
        language translators.

        LTESID  If CD number is:
                zero: The ESID number was not yet processed.
                      Returns to the address in the link
                      register.
                negative: The ESD record is bypassed. Returns to
                      link register + 4. Addresses of the
                      linkage table entry and the control
                      dictionary number are supplied.
                positive: Returns to register + 8. Relocation
                      factor for SD/PC, control-dictionary
                      -number and -address are supplied.
```

Subr.	Entry	Function
SRCHCD	SRCHCD	Searches the CD for a matching label.
	SRPCOD	Continues the search after a matching label has been found.
CNVHEX	CNVHEX	Converts EBCDIC input into hexadecimal output.
PRINT	PRINT	Prints messages and link map on SYSLST.
LOGMSG	LOGMSG	Prints error message on SYSLOG.
PRTLST	PRTLST	Prepares for printing the linkage editor diagnostics of input.
SPACE1	SPACE1	Spaces one line on SYSLST.
XTPHNO	XTPHNO	Extracts the phase number from CD entries for SD, PC, LD, or LR records.
	XTPHGT	Entry XTPHGT is used if the entry is known to be an SD or PC.
ABTERR	ABTERR	Gives control to --------------------> IJBRLD for handling of an abnormal termination error.
CDSIZE	CDSIZE	Checks for CD overflow.
ALNKPR	ALNKPR	Initializes for the scanning of the sublibrary member index for Autolink. Extracts unresolved ERs from the CD in collating sequence and includes the corresponding modules.
ALNKOF	ALNKOF) RDNEXT)	Reads the input stream and diagnoses the type of statement to pass control to the appropriate CSECT.
	RDEXEC) EXLOAD)	Entry at RDEXEC skips reading of statement.
CANCL	CANCL	Cancel routine.
ERROR	ERROR	Sets up to print non-termination error messages. If the calling routine sets the RETRN bit in ERRSW, returns to caller. If RETRN is off, returns to RDNEXT or ALNKPR if error during Autolink.
NOTCTL	NOTCTL	Converts input statements (X'02' in first byte) to print format.

CSECT IJBOTH

Called By: IJBLNK

```
IJBOTH : - Initializes IJBOTH.
         - Branches to the right processor for the statement at hand:
           ---------->TXTPRC
           ---------->REPROC
           ---------->RLDPRC
           ---------->ENDPRC
TXTPRC : TXT processor
         - Puts text into CATALOG sublibrary format.
         - Calls INLPLEIT to write text. -------> RDNEXT (IJBLNK)
REPPRC : REP processor
         - Modifies REP to text.
         - Processes in TXTPRC. ----------------> RDNEXT (IJBLNK)
RLDPRC : RLD processor
         - Converts R and P pointer information to CD
           number information for RLD pass 2.
         - Stores RLD records in buffer (writes to IJSYS01
           if necessary.) ---------------------> RDNEXT (IJBLNK)
ENDPRC : END processor
         - Updates input control mechanism (LIFO stack PERIDA).
         - Supplies transfer address.
         - Identifies unassigned LD/LR in control
           dictionary for this module.
         - Clears the linkage table.
         - Accepts CSECT length if necessary. (Some
           language translators supply CSECT length
           in end record.)
         - If Autolink necessary
           goes to -----------------> ALNKPR (IJBLNK)
           Else goes to -------------> RDNEXT (IJBLNK)
```

CSECT IJBFIN

Called By: IJBLNK

```
READIN  : - Controls record I/O.
LNKPOINT: - Points to next record.
LNKNOT  : - Notes record.
LCFIND  : - Finds module.
IJBLETR : - Handles messages.
LEITCALL: - Calls INLPLEIT for librarian services.
LCLOSE  : - Disconnects all connected libraries and sublibraries.
```

CSECT IJBESD

Called By: IJBLNK

```
  IJBESD : - Controls input for updating LIFO-stack PERIDA.
           - Builds dummy PHASE statements if none supplied.
  ESDRET : - Checks validity of type code for ESD record.
           - Exits if all ESD records are processed.
           - If end-of-record or SYM record found
             goes to ------------------> RDNEXT (IJBLNK)
             Else preliminary processing of SD/PC/LD/ER/CM records.
           - Compares label on input ESD with CD.
           - Completes SD/PC processing by ensuring that
             name field on ESD record is blank.
           - Completes processing:
  ELBCM  :   - CM
  ELBSD  :   - SD/PC
  ELBLD  :   - LD
  ELBER  :   - ER
           - If no dictionary update ---------> ESDRET
           - Else:
             -- Posts ESD record in CD.
             -- Moves CD number to linkage table.
             -- Checks for CD and linkage table overflow.
             -- Goes to ----------------------> ESDRET
```

In detail, ESD processing takes the following steps:

Pre-Processing

1. For each ESD record produced by a language translator, an input CD entry is built at a fixed location in storage. In some cases, this input CD entry is moved to the CD during processing.

2. The input ESD type field is validated.

   - If it is a weak external, the ESD type field in the input CD entry is set to ER and the NOAUTOL and WXTRN bits in CSWITCH are turned on.

   - If it is invalid, an error condition exists, the whole ESD record is ignored, and the next ESD record is processed.

3. Further preprocessing depends on the ESD type:

   - For LD input: An LD record has a pointer to the linkage table where the CD number is checked to see if the LD record has already been processed.

   - For ER input: If NOAUTO was specified, the NOAUTOD bit in CSWITCH is set on.

- For SD or PC input: Two conditions must be fulfilled:

  a. The assembled origin must be aligned on a double-word boundary.

  b. The PC must be unnamed.

  The relocation factor is calculated by subtracting the assembled origin from the storage address (NXPHRG).

For a normal INCLUDE, pre-processing is finished at this stage.

For a submodular INCLUDE, the name list of included CSECTs is scanned for a name identical to the name of the input CD entry. If the names match, pre-processing is finished. If not, the ESD type field in the input CD entry is changed to ER and a switch is set to ensure that the CD number in the linkage table is given a negative value.

Processing

1. The CSECT scans the CD for an entry with the same name as the input ESD.

   This scan starts at the end of the CD and proceeds towards the beginning until either a match occurs or the beginning of the CD is reached. If a match occurs, the CD entry is called a duplicate.

   The scan continues if the duplicate is a phase entry.

2. If no duplicate is found, the input CD entry is added to the end of the CD.

3. If the input ESD is an SD, PC, CM, or ER, an entry is made in the linkage table.

4. If a duplicate is found, the action taken by the ESD processor depends on the relationship between input and duplicate. Figure 6 shows all possible actions and their abbreviations A1 to Err-46.

Action	Meaning
A1:	Ignores input CD entry.
A2:	Adds input CD entry to the end of CD.
A3:	Replaces duplicate with the input CD entry.
A4*:	Adds the linkage table entry pointing to the last entry added to the CD.
A5*:	Adds the linkage table entry pointing to the duplicate.
A6:	Changes duplicate LD to LR.
A7:	Continues scan of CD.
A8:	Saves length of longest CM in CD.
A9:	Gives CD number in linkage table a negative value.
A10:	Changes input LD to LR.
A11:	Sets 'Possible Duplicate Entry' switch.
Err-43:	Prints error message '2143I' and goes to RDNEXT.
Err-46:	Prints error message '2146I' and goes to RDNEXT.

----

*If a submodular INCLUDE was used and the name list of included SDs does not contain an SD, the CD number in the linkage table is given a negative value.

Figure 6. ESD Processing Actions

To find which action is taken while processing input CM, ER, SD, or LD records, use Figures 7 to 10. The upper part of these figures shows the various conditions which exist (Y), do not exist (N), or can be ignored (-), while the lower part indicates the actions taken (X).

Duplicate = SD	Y	N	N	N	N
= PC	N	Y	N	N	N
= CM	N	N	Y	N	N
= LD/LR	N	N	N	N	Y
= ER	N	N	N	Y	N
A3	-	-	-	X	-
A4	-	-	-	-	-
A5	X	-	X	X	-
A7	-	X	-	-	-
A8	-	-	X	-	-
Err-46	-	-	-	-	X

Figure 7. Decision Table if Input is a CM Record

Duplicate = SD, LD, or LR	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	Y
= LD	-	-	N	N	N	N	Y	Y	Y	Y	N	N	N	N	-
= CM	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N
= ER	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	N
Duplicate unassigned *	N	N	N	N	N	N	N	N	N	N	-	-	-	-	Y
Name = 'IJ..' or 'IBM..'	Y	Y	Y	Y	Y	N	Y	Y	Y	N	-	-	-	-	-
Name = 'IBM..'	N	Y	-	-	N	-	-	-	N	-	-	-	-	-	-
NOAUTO for input	N	N	Y	N	N	-	Y	N	N	-	-	-	Y	N	-
Duplicate in current phase	N	N	-	Y	N	-	-	Y	N	-	-	N	Y	Y	-
Duplicate in ROOT phase	N	-	-	-	Y	-	-	-	Y	-	-	-	-	-	-
A2	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-
A3	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-
A4	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-
A5	-	-	X	X	X	X	X	X	X	X	X	X	X	X	-
A6	-	-	-	-	-	-	X	X	X	X	-	-	-	-	-
A7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X

* SD is to be considered assigned

Weak externals are processed like ERs for which NOAUTO is requested.

Figure 8. Decision Table if Input is an ER Record

Duplicate = SD	N	N	N	N	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N
= CM	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
= LD or LR	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
= ER	N	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Duplicate unassigned	-	-	-	-	-	-	-	-	N	N	N	Y	Y	Y	Y	Y	Y	Y
Input and dupl. ESIDs agree	-	-	-	-	-	-	-	-	-	-	-	N	N	N	Y	Y	Y	Y
ASSORGs agree	-	-	-	-	-	-	-	-	-	-	-	-	-	-	N	N	N	Y
Duplicate in current phase	-	-	Y	N	Y	N	N	N	-	Y	N	-	Y	N	-	Y	N	-
Name = 'IBM..'	-	N	Y	Y	-	Y	N	N	N	Y	Y	N	Y	Y	N	Y	Y	-
Duplicate in ROOT phase	-	-	-	-	-	-	N	Y	-	-	-	-	-	-	-	-	-	-
A2	-	-	-	X	-	X	X	-	-	-	X	-	-	X	-	-	X	-
A3	X	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X
A4	-	-	-	X	-	X	X	-	-	-	X	-	-	X	-	-	X	-
A5	X	X	X	-	X	-	-	X	-	-	-	-	-	-	-	-	-	X
A9	-	-	-	-	X	-	-	X	-	-	-	-	-	-	-	-	-	-
Err-43	-	-	-	-	-	-	-	-	X	X	-	X	X	-	X	X	-	-

Figure 9. Decision Table if Input is an SD Record

Duplicate = SD, LD, or LR	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
= CM	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
= LD or LR	N	N	N	N	-	-	-	-	-	Y	Y	Y	Y	Y	Y	Y	N	N	N	N
= ER	N	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Duplicate unassigned **	-	-	-	-	N	N	N	N	N	N	N	N	N	N	N	Y	-	-	-	-
Input unassigned	-	-	-	-	-	-	-	Y	Y	N	N	N	N	N	N	-	N	N	N	N
Input points to duplicate	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Y	N	N	N
Input and duplicate point to the same entry	-	-	-	-	-	-	-	-	-	Y	N	N	N	N	N	-	-	-	-	-
Names of C/D entries agree	-	-	-	-	-	-	-	-	-	-	N	N	N	Y	Y	-	-	-	-	-
Name of input and duplicate = 'IBM...'	-	N	Y	Y	N	Y	Y	-	-	-	N	Y	Y	-	-	-	-	N	Y	Y
Duplicate in current phase	-	-	Y	N	-	N	Y	N	Y	-	-	N	Y	N	Y	-	-	-	N	Y
Input and duplicate ASSORGs agree	-	-	-	-	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	Y
A1	-	-	-	-	-	-	-	-	X	X	-	-	-	-	X	-	X	-	-	-
A2	-	-	-	X	-	X	-	X	-	-	-	X	-	X	-	-	-	-	X	-
A3*	-	X	X	-	-	-	-	-	-	-	-	-	-	-	-	X*	-	-	-	-
A10	-	X	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A11	-	-	-	-	-	-	-	-	X	-	-	-	-	-	X	X	-	-	-	-
Err-43	-	-	-	-	X	-	X	-	-	-	X	-	X	-	-	-	-	X	-	X
Err-46	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

* Action A3 is performed retaining the ESD type of the duplicate
** SD is to be considered assigned

Figure 10. Decision Table if Input is an LD Record

Post-Processing

1. For ER, LD/LR, or CM input, the next ESD record is selected for processing.

2. For SD or PC input

   a. The CD is scanned for unassigned LDs or LRs pointing to the input record.

   b. The CD entries found during the previous scan are updated. This is done by storing in the CD entry the CD number found in the linkage table entry that corresponds to the input item.

   c. The storage address (NXPHRG) is updated by adding the length of the CSECT.

If the length of the CSECT is provided in the END statement, CSECT IJBOTH performs action c.

CSECT IJBSCN

```
Called By: IJBLNK

  IJBSCN : - Controls input for updating PERIDA.
           - Finds operation field and checks validity.
           - Branches to the processor for this type of statement:
             ------------> INCCRD
             ------------> PHCRD
             ------------> ENTCRD

  INCCRD : INCLUDE statement processor
           - Checks validity of operands.
           - If no operand, -----------------> RDNEXT (IJBLNK)
           - Else locates module to be included and autolinks.
           - Goes to -----------------------> RDNEXT (IJBLNK)

  PHCRD  : PHASE statement processor
           - Checks validity of operands.
           - If Autolink required -----------> ALNKPR (IJBLNK)
           - Else goes to -------------------> IJBCTL

  ENTCRD : ENTRY statement processor
           - Provides exit from ENTRY or PHASE statement processors.
           - Saves transfer address if in ENTRY statement.
           - If Autolink --------------------> ALNKPR (IJBLNK)
           - Else goes to -------------------> IJBCTL
```

CSECT IJBCTL

```
Called By: IJBSCN

  IJBCTL : - If first phase -------------------> PHSPRC
           - Else ----------------------------> WRTRFR

  WRTRFR : PHASE post-processor
           - Reserves space for relocation information.
           - If Autolink ---------------------> ALNKPR (IJBLNK)
           - Else
             -- If PHASE statement ------------> PHSPRC
             -- If ENTRY statement ------------> IJBMAP

  PHSPRC : PHASE pre-processor
           - Determines optional operands specified.
           - Builds current phase CD entry.
           - Gets information to processing phase.
           - Determines if relocation possible.
           - Goes to --------------------------> RDNEXT (IJBLNK)
```

CSECT IJBMAP

```
            Called By: IJBCTL

              IJBMAP : - Displaces phase load address by cumulative length
                         of commons.
                       - Calculates load origin for transfer address.
                       - Sorts CSECTs by load address.
                       - If option SMAP, sorts CSECT names alphabetically
                         and produces list.
                       - If option MAP prints map.
                       - Exits depending on errors and option CANCEL.
                         Goes to ----------------> EOJ or IJBRLD
```

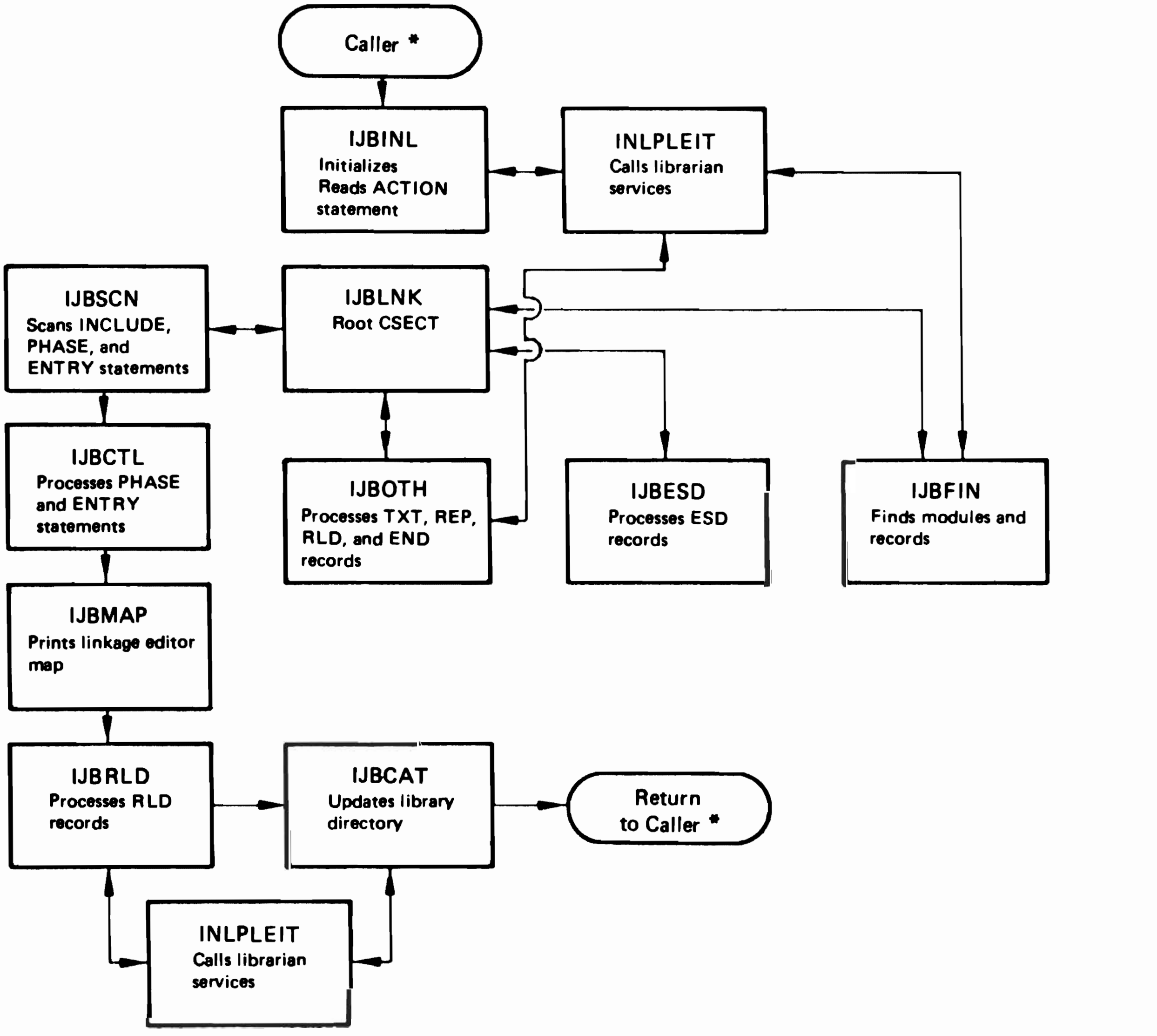
CSECT IJBRLD

```
            Called By: IJBMAP

              IJBRLD :  - If no more RLDs --------> TSTUNR (IJBRLD)
                        Pass2 P-pointer processor
                         -- Reads RLDs.
                         -- Gets relocation factor for P-pointer.
                         -- Calls INLPLEIT to move the correspondent CD entry
                            to the current phase entry, if the P-pointer is
                            outside the current phase.
                         -- Control flow in IJBRLD depends on conditions found.
              RLDOR  :  Pass2 R-pointer processor
                         -- Gets relocation factor for R-pointer.
                         -- Adds the assembled origin (i.e. the address of the SD,
                            PC or CM that defines the ER) if R-pointer is an ER.
                         -- If a constant must not be processed ---> IJBRLD
              RLDCON : Pass2 RLD constant processor
                       - Adjusts constant portion of RLD record by
                         relocation factor.
                       - Calls INLPLEIT to update ADCONs in phase
                         text in CATALOG sublibrary ----------> IJBRLD
              TSTUNR   - If RLD PASS 3 ----------------------> WRLST (IJBRLD)
                       - If MAP option ----------------------> TSTCNT
              TSTCNT : MAP routine
                       - Lists unresolved ADCONs.
                       - Lists ADCONS outside of limits of current phase.
                       - If phase not relocatable ------------> WRLST (IJBRLD)
                       - Initializes Pass3 RLD processing.
                         -- Modifies Pass2 RLD processor to insert relocation
                            information for relocatable phases in space
                            reserved by WRTRFR (IJBCTL).
                         -- Positions at start of RLD information.
                       - Adds RLD information to phase text.
              WRLST  : Block phase header
                       - Calls INLPLEIT to write last buffer to library.
                       - Closes SYSLNK and IJSYS01 and --------> IJBCAT
```

CSECT IJBCAT

```
Called By: IJBRLD

  IJBCAT : - Starts stow table.
           - Initializes CD search.
  SCANCD : - Locates next phase entry in CD.
           - If end of stow table:
             - Calls INLPLEIT to update sublibrary member index or,
               in case of a previous severe error, deletes the
               phases already written into the CATALOG sublibrary.
             - Starts new stow table.
           - Modifies load and transfer address of phase in process.
           - Adds entry to stow table.
           - If more entries in CD, --------------------------> SCANCD
           - Calls INLPLEIT to disconnect any CATALOG sublibrary
             or SEARCH sublibrary connected.
           - Sets return code into MSHPRET:
               0 if successful link
               4 if warning or error issued, but phases are cataloged
              16 if severe error
           - Returns to caller (MSHEOJ)
```

Figure 11 shows how the CSECTs of the program are given control during a linkage editor run.



*)The caller can actually be job control or MSHP.

Figure 11. Control Flow

DATA AREAS

LIBRARY RECORD FORMATS

The linkage editor input records coming from the library have the format of library members of type OBJ. The records for output have the format of library members of type PHASE.

Input Record Formats

The input records for the linkage editor are in card image format:

The ESD Record

Card Columns

1 Hex 02; identifies a loader record.
2 - 4 ESD record
11 - 12 Number of bytes of information contained in this record
15 - 16 ESID number of the first SD, PC,CM, or ER on this record. Relates it to a CSECT.
17 - 72 Variable information:
8 positions - Name
1 position - Type code hex 00, 01, 02, 04, 05, or 0A, to indicate SD, LD, ER, PC, CM, or WX.
3 positions - Assembled origin
1 position - Blank
3 positions - Length if SD, CM, or PC. If LD, ESID number of SD containing the label.

73 - 80 May be used by the programmer for identification.

The TXT Record

Card Columns

1 Hex 02; identifies a loader record.
2 - 4 TXT record
6 - 8 Assembled origin (address of first byte to be loaded from this record)
11 - 12 Number of bytes of text to be loaded
15 - 16 ESID number of the CSECT (SD or PC) containing the text
17 - 72 Up to 56 bytes of text -- data or instructions to be loaded
73 - 80 May be used for program identification.

The RLD Record

Card Columns

```
 1      Hex 02; identifies a loader record.
 2 -  4 RLD record
11 - 12 Number of bytes of information contained in this record.
17 - 72 Variable information:
  2 positions -  R-pointer:
                 pointer to the ESID number of the ESD on which the
                 relocation factor or the contents of the ADCON depends.
  2 positions -  P-pointer:
                 pointer to the ESID number of the ESD on which the
                 position of the ADCON depends.
  1 position  -  flag bits indicating type of constant:
       0 - 2  ignored
       3      0 - a non-branch type load constant
              1 - a branch type load constant
       4 - 5 00 - load constant length = 1 byte
             01 - load constant length = 2 bytes
             10 - load constant length = 3 bytes
             11 - load constant length = 4 bytes
       6      0 - relocation factor is to be added
              1 - relocation factor is to be substracted
       7      0 - Next load constant has different R- and P-pointers.
                  R and P must be present.
              1 - Next load constant has the same R- and P-pointers.
                  Therefore they are both omitted.

  3 positions  -  assembled origin of load constant.

73 - 80 May be used for program identification.
```

The END Record

Card Columns

```
 1      Hex 02; identifies a loader record.
 2 -  4 END record
 6 -  8 Assembled origin of the label supplied to the assembler in the
        END record (optional)
15 - 16 ESID number of the CSECT to which this END record refers.
        Only if 6-8 present
17 - 22 Symbolic label supplied to the assembler if this label was not
        defined within the assembly
29 - 32 CSECT length (if not specified in last SD or PC)
73 - 80 Not used
```

The REP Record

Card Columns

1 Hex 02; identifies a loader record.

2 - 4 REP record

5 - 6 Blank

7 - 12 Assembled hex address of the first byte to be replaced. Right justified with leading zeros. (Note that there is no check to determine if the the assembled address is actually within this CSECT)

13 Blank

14 - 16 ESID hex number of the CSECT (SD) containing the text. Right justified with leading zeros.

17 - 70 From 1 to 11 4-digit hex fields separated by commas. Each field takes two bytes. A blank shows end of information.

71 - 72 Blank

73 - 80 May be used for program identification

Object Member Record Format

In a sublibrary, an object module has only records of 80 bytes. These records contain either one and only one linkage editor statement or one of the five possible types of module record types (ESD, TXT, RLD, REP, or END).

When job control or a language translator writes them on SYSLNK it adds to them two bytes of control information, a blocking factor and the record length of 80 bytes, and possibly blocks them to a block length of 322 bytes.

Figure 12 shows the two possible record formats on SYSLNK.

```
            |<--82 Bytes--->|
            |--+--+---------|
Format 1    |B1|B2| 80 Bytes|
            |__|__|_________|

            |<------------322 Bytes-------------------->|
            |--+--+---------+---------+---------+---------|
Format 2    |B1|B2| 80 Bytes| 80 Bytes| 80 Bytes| 80 Bytes|
            |__|__|_________|_________|_________|_________|

B1(1 byte)  -  Blocking factor (either 1 or 4)
B2(1 byte)  -  Record Length = 80
```

Figure 12. OBJ-Member Record Formats on SYSLNK

Figure 13 shows the format of a complete linkage editor input module on SYSLNK. The linkage editor statements may already be stored with the module in the sublibrary.

4	80	INCLUDE	INCLUDE	PHASE	ESD
4	80	ESD	ESD	TXT	TXT
4	80	TXT	TXT	TXT	TXT
4	80	TXT	TXT	TXT	REP
4	80	REP	RLD	RLD	RLD
4	80	RLD	END		

<-->

322 Bytes

Figure 13. Example of a Module on SYSLNK

SYSLNK Control Interval Format

Figure 14 shows a VSAM or FBA control interval (CI) with SYSLNK input records for the linkage editor.

```
<----------------------Control Interval------------------------->
┌────────────────┬─────────────────┬──────┬───────┬───────┬───────┐
│record of 82 or │ record of 82 or │ ...  │ RDF1  │ RDF2  │ CIDF  │
│   322 bytes    │    322 bytes    │      │3 bytes│3 bytes│4 bytes│
└────────────────┴─────────────────┴──────┴───────┴───────┴───────┘

CIDF: control interval definition field;
      bytes 1,2: address of free space
      bytes 3,4: length of free space

RDF1: record definition field;
      byte 1: bit 1=0: only one 512 byte block in CI.
              bit 1=1: more than one 512 byte block in CI.
      bytes 2,3: length of block (512 bytes)

RDF2: bytes 2,3: number of 512 byte blocks on this CI.
```

Figure 14. VSAM or FBA Format of SYSLNK Records

Output Record Formats

A phase is stored into the CATALOG sublibrary as a member of type PHASE. It contains executable code and, at the end of the phase, some RLD information, if the phase is relocatable.

RLD Block Format

The RLD items stored by CSECT IJBOTH in the RLD buffer are blocked to a length of 2020 bytes. If the RLD buffer is full, its content is written on workfile IJSYS01.

CSECT IJBRLD places the RLD items into the CATALOG sublibrary. The layout of an RLD block is shown in Figure 15.

Offset	Length	Content
0	4	Number of RLD items in one block
4	2016	RLD items

Figure 15. Layout of an RLD Block

LINKAGE EDITOR DATA AREAS

Control Dictionary (CD)

FUNCTION: The CD holds information on phases and modules for the address adjustment of the linkage editor.

NAME: Control Dictionary

LABEL OR IDENTIFIER: None

LOCATION: Partition

INITIALIZED BY: IJBINL

POINTED TO BY: CDENT1

USED BY: All CSECTs except IJBINL and INLPLEIT

FORMAT OF CD ENTRIES: The CD has entries for ESD information, for modules, and for phases. The entries for phases and for ESD records are first built one by one in CSECT IJBLNK or stored temporarily in the current CD entries. During processing, the references to CD information are sometimes to fields of such "current entries" in the CSECTs instead of the CD entry field. In IJBLNK, a current phase entry is stored under the label CPHENT and a current ESD entry is stored under the label CESDENT. The layout of CD entries is the following:

OFFSETS		TYPE	LENGTH	NAME	DESCRIPTION
0	(0)	STRUCTURE	0	CDENTRY	
0	(0)	SIGNED	4		
ENTRY FOR SD,PC,CM,LD,LR,ER,WX RECORDS					
0	(0)	CHARACTER	8	NAMED	NAME OF ESD ITEM
8	(8)	HEX	1	ESDTYPD	ESD TYPE
9	(9)	HEX	3	ASSORGD	ASSEMBLED ORIGIN OR
9	(9)	HEX	3	LNGTHD	LENGTH OF CM
12	(C)	SIGNED	4	RELFACD	RELOCATION FACTOR
12	(C)	SIGNED	2		
14	(E)	SIGNED	2	CSNUMD	OR C/D # OR ESID # OF CSECT FOR LD/LR
14	(E)	SIGNED	2	PHNUMED	OR PHASE # FOR ER/WX
16	(10)	SIGNED	2	PHNUMD	PHASE # FOR SD/PC
18	(12)	HEX	1	SWITCHD	DIVERSE SWITCHES
19	(13)	HEX	1	SWITCHD2	ADDITIONAL FLAG BYTE
20	(14)	SIGNED	2	SDXREFD	XREF TO INPUT MODULE
22	(16)	SIGNED	2		
ENTRY FOR PHASE					
0	(0)	CHARACTER	8	PHNAMED	PHASE NAME
8	(8)	HEX	1	ESDTYPED	ESD TYPE
9	(9)	HEX	5		RESERVED
14	(E)	SIGNED	2	RLDITEMD	# OF RLD ITEMS
16	(10)	SIGNED	4	TXTLEND	TEXT LENGTH IN BYTES
20	(14)	SIGNED	4	RLDLRBD	LRBA FOR RLD ITEMS
24	(18)	SIGNED	4	ORPHRGD	PHASE ORIGIN
28	(1C)	SIGNED	4	NXPHRGD	NEXT PHASE ORIGIN
32	(20)	SIGNED	4	TRFRADD	TRANSFER ADDRESS
36	(24)	SIGNED	4	LINKSTRD	START OF PARTIT.
40	(28)	SIGNED	2		NUMBER OF EXTRA RLD BLOCKS
42	(2A)	CHARACTER	1	PHTYPED	PHASE TYPE
43	(2B)	CHARACTER	1	PHTYP2D	ADDITIONAL FLAGS
1...				PHTYP2R	ROOT PHASE
.1..				PHTYP2T	FIRST PHASE PROCESSED
..1.				PHDUMMY	PHASE ENTRY WITHOUT TEXT
44	(2C)	SIGNED	4	DENTXREF	XREF TO CORRESP. DICT.ENTRY
ENTRY FOR MODULE NAME					
0	(0)	CHARACTER	6		
6	(6)	CHARACTER	2	LCDFLG	RESERVED
8	(8)	HEX	1	LCDMTYP	TYPE
9	(9)	HEX	1	LCDMVER	VERSION
10	(A)	HEX	1	LCDMMOD	MODIFICATION LEVEL
11	(B)	HEX	1		
12	(C)	CHARACTER	8	LCDMNAM	MODULE NAME
20	(14)	SIGNED	4		

```
MASKS FOR ESDTYPD / ESDTYPED
  .... ....      SD   SECTION DEFINITION
  .... ...1      LD   LABEL DEFINITION
  .... ..1.      ER   EXTERNAL REFERENCE
  .... ..11      LR   LABEL REFERENCE
  .... .1..      PC   PRIVATE CODE
  .... .1.1      CM   COMMON
  .... .111      PH   PHASE ENTRY
  .... 1.1.      WX   WEAK EXTERNAL
  ...1 ....      IC   INCLUDED MODULE NAME
  ..1. ....      DE   SAVED DIRECTORY ENTRY

MASK FOR SWITCHD
  .... ...1     UNASSG
  .... ..1.     WXTRN   THE ER IS WEAK EXTERNAL
  .... .1..     NOAUTOL NO AUTOLINK NECESSARY
  1111 111.     ASSG    MASK TO ASSIGN LD/LR

MASKS FOR PHTYPED
  1... ....     SELFRELO  SELFRELOCATING PHASE
  .1.. ....     RELPHASE  RELOCATABLE PHASE
  ..1. ....     SVAELIG   SVA ELIGIBLE
EQU   X'00'               NOT RELOCATABLE
```

Linkage Table

FUNCTION: The language translator gives each ESD record a number called ESID number. The linkage editor gives it a CD number unique in the phase, because the same ESID number might occur several times coming from the different modules. A linkage table links the ESID number of each record from the language translator to the CD number given to the record by the linkage editor.

NAME: Linkage Table

LABEL OR IDENTIFIER: None

LOCATION: Partition

CHANGED BY: IJBESD and IJBCTL

USED BY: All except INLPLEIT

POINTED TO BY: LTMINE (= address of first item in linkage table minus 3)

LAYOUT: A linkage table has up to 511 3-byte entries. Figure 16 shows the format of an entry. Each object module has its own linkage table. When an END record is processed, signalling the end of a module, the table is reset to zeros.

Control Dictionary Number	ESD Type
2 Bytes	1 Byte

Figure 16. Linkage Table Entry Format

LIFO Stack (PERIDA)

FUNCTION: The LIFO stack is built when an INCLUDE statement is processed and used to

1. obtain the address of the next record after the END statement
2. determine the end of processing for an object module
3. control the nesting of INCLUDE statements to give priorities.

NAME: LIFO stack

LABEL: PERIDA

LOCATION: Top in IJBLNK

INITIALIZED BY: IJBINL

CHANGED BY: IJBCTL

POINTED TO BY: LCSTBEG

LAYOUT:

Label	Offset	Length	Contents
PERIDA	0	6	NOTE information for SYSLNK records
PERISW	6	1	Status information: X'01' SYSLNK input X'02' Named submodular X'08' Autolink active for current module X'20' SYSLNK on FBA or VSAM managed space X'80' Library input of type OBJ
PERILRC	7	1	Record count within block
PERIRRN	8	4	Relative record count
PERIMNO	C	2	Cross reference to module name for CSECTs being linked (CD offset)
PERIRL	E	1E	NOTE information for relocatable modules

Figure 17. Layout of the LIFO Area

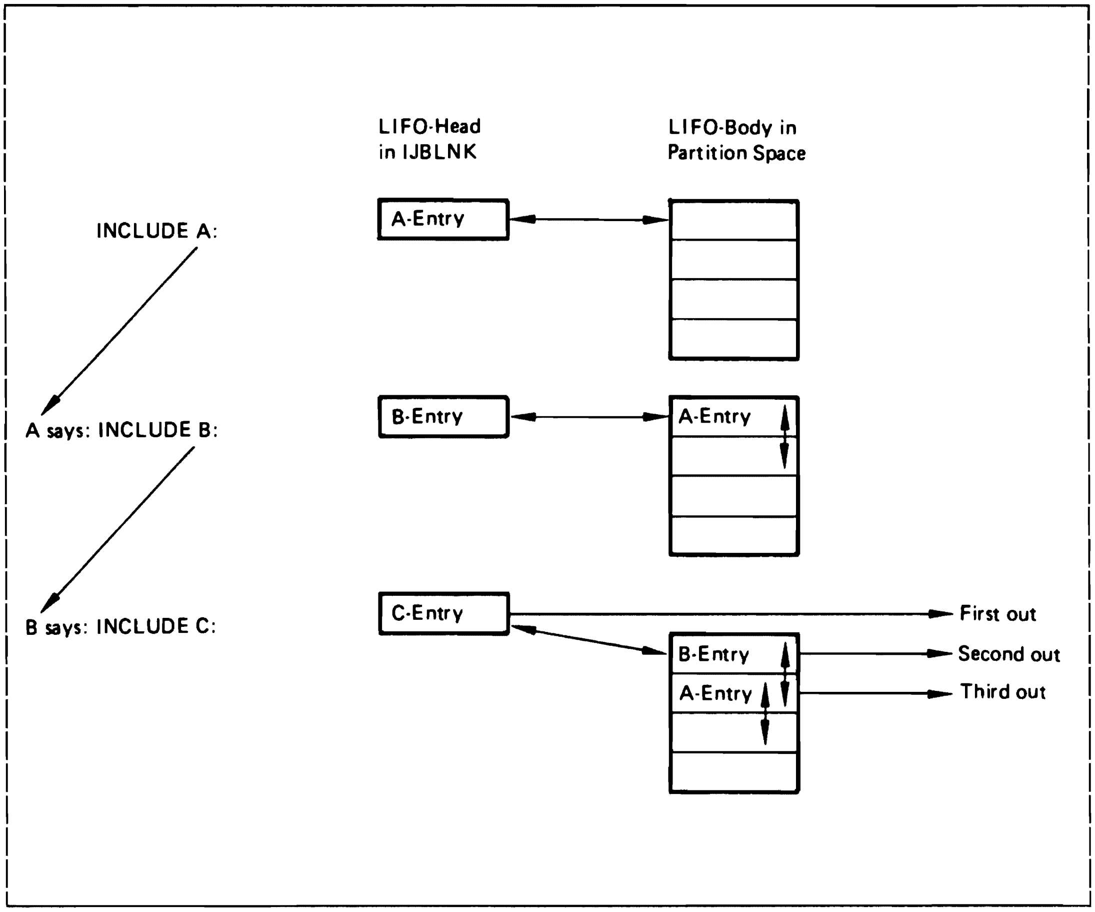


Figure 18. How the LIFO Stack Works

Communication Area LEITPL

FUNCTION: Communication area between module (and CSECT) INLPLEIT and the CSECTs of module IJBLE1.

NAME: LEITPL

LABEL: LEITPL

LOCATION: CSECT IJBLNK

CHANGED BY: All CSECTs of module IJBLE1.

LAYOUT:

OFFSETS		TYPE	LENGTH	NAME	DESCRIPTION
0	(0)	STRUCTURE	0	LEITPL	
0	(0)	SIGNED	4		
LINKAGE EDITOR AND LIBRARIAN SERVICES INTERFACE					
0	(0)	V-ADDRESS	4	LEITA	"V(INLPLEIT)" ENTRY ADDR TO LE INTERFACE MODULE
4	(4)	SIGNED	4	LEITPL	LE INTERFACE PARM LIST
4	(4)	HEX	1	LEITFC	FUNCTION CODE
5	(5)	HEX	1	LEITSW	SWITCHES
		1...		LEITCHK	CHECK ON RECORD REQUEST
		.1..		LEITMOLD	CONNECT USING OLD MACB
		..1.		LEITLINK	LINK OPTION
		...1		LEITMSCN	RUN UNDER MSHP CNTL
		.... 1...		LEITMSPA	PHASE UNDER MSHP CNTL
		.... .1..		LEITMSBY	MSHP BYPASS REQUEST
		.... ..1.		LEITMS1T	1TH TIME MSHP
		1		LEITMFND	PHASE ALREADY IN LIBRARY
6	(6)	HEX	1	LEITSW1	
		1...		LEITNOCA	NO CATALOG SUBLIB GIVEN
		.1..		LEITST1T	1ST TIME STOW
		..1.		LEITEOM	END OF OBJ MODULE
		...1		LEITSEVE	SEVERE ERROR OCCURED
		.... 1...		LEITFULL	LIBRARY FULL
		.... .1..		LEITSDEL	FORCE STOW DELETE
		.... ..1.		LEITGVIS	LAMB AVAILABLE FOR GETVIS
		1		LEITCCON	CATALOG LIBRARY CONNECTED
7	(7)	HEX	1	LEITSW2	
		1...		LEITDCAN	DELAYED CANCEL FORCED
		.1..		LEITCARQ	CANCEL REQUEST PENDING
		..1.		LEITSCON	A SEARCH SUBLIBRARY CONNECTED
8	(8)	HEX	8	LEITSBUF	SBUF DESCRIPTOR
16	(10)	HEX	8	LEITPBUF	PBUF DESCRIPTOR
24	(18)	A-ADDRESS	4	LEITCOMM	COMMUN. AREA PTR
28	(1C)	CHARACTER	8	LEITPNAM	PHASE NAME
36	(24)	SIGNED	4	LEITMARK	HI WATER MARK
40	(28)	SIGNED	4	LEITOFFS	OFFSET FOR PUT
44	(2C)	A-ADDRESS	4	LEITTXT	ADDR. OF TEXT
48	(30)	SIGNED	4	LEITTXTL	LENGTH OF TEXT
52	(34)	A-ADDRESS	4	LEITSTOW	ADDR OF STOW LIST
56	(38)	HEX	1		
57	(39)	HEX	1	LEITRC	RETURN CODE
58	(3A)	HEX	1		
59	(3B)	HEX	1	LEITEC	ERROR CODE
60	(3C)	SIGNED	4	LEITVIOO	PHASE START ADDRESS
64	(40)	A-ADDRESS	4	LEITDTF	ADDR. OF LIST DTF
68	(44)	A-ADDRESS	4	LEITIO	ADDR. OF DTF I/O AREA
72	(48)	HEX	168	LEITARG	LENGTH IS MULTIPLE OF DIR.ENTRY FOR STOW CD FIELDS

FUNCTION CODES IN LEITFC

....1	OPENCAT
.... ..1.	NEWMEM
.... ..11	DISCMEM
.... .1..	DISCLIB
.... .1.1	PUTREC
.... .11.	GETREC
.... .111	STOW
.... 1...	FIND
.... 1..1	NOTE
.... 1.1.	POINT
.... 1.11	UPDTE
.... 11..	OPENOBJ
.... 111.	FINISH

ERROR CODES IN LEITEC

...1 11..	NOCATSL	NO CATALOG SUBLIB
..1.	VIOEX	VIO EXHAUSTED
..11	MSHPCNPA	PHASE CANNOT BE REPLACED
.1..	LIBFULL	LIBRARY IS FULL

LIBRARIAN DATA AREAS USED BY THE LINKAGE EDITOR

Stow Table

FUNCTION: In this table the directory information for all phases produced during a linkage editor run is collected for the sublibrary directory.

NAME: Stow Table

LABEL: None

POINTED TO BY: LTMINE in IJBLNK

INITIALIZED BY: IJBCAT

CHANGED BY: IJBCAT

LAYOUT: See VSE/Advanced Functions Diagnosis Reference: Librarian LY33-9111, data area INLCDENT.

Buffers for Librarian Services

FUNCTION: The linkage editor provides these buffers for the librarian services it requests:
The S-buffer is a shared buffer for library management of the CATALOG sublibrary. The P-buffers are used for member I/O, one to put phases into the CATALOG sublibrary and one to read input (modules or phases) from OBJ-search chain sublibraries.

NAMES: S-Buffer and P-Buffers

LABELS: None

POINTED TO BY: SHARBUF, PHAPBUF, OBJPBUF in IJBLNK

LOCATION: Partition of linkage editor

ALLOCATED BY: IJBINL

FORMATTED BY: Librarian

CHANGED BY: Librarian

LAYOUT: See VSE/Advanced Functions Diagnosis Reference: Librarian LY33-9111, data areas INLCBUCB and INLCBHDR.

DIAGNOSTICS

LINKAGE EDITOR LISTING AND MAP

The linkage editor program lists for each execution the errors encountered and actions taken. This listing is followed by a linkage editor map prepared by CSECT IJBMAP.
For a detailed description of the linkage editor listing and map see VSE/Advanced Functions: Service Aids SC33-6195.

The linkage editor map has the following columns:

1. Phase name
2. Transfer address
3. Start and end of the virtual storage location
4. For each CSECT of the phase:
   - labels in ascending order
   - load address
   - relocation factor
   - offset of CSECT in the partition where link-edited
   - offset of CSECT in phase
   - name of object module from which the CSECT was taken (or SYSLNK)

Figure 19 shows an example of a linkage editor map.

```
┌------------------------------------------------------------------------->
|PHASE    XFR-AD  LOCORE  HICORE  CSECT/   LOADED  RELOC. PARTIT...
|                                 ENTRY    AT      FACTOR OFFSET
|ASSEMBLY 03C282  03A078  03CC40
|                                 IPKAJ000 03A078  03A078 000000
|                                 IPKAA002 03A080  039600 000008
|                                 IPKAA000 03AA38  03AA38 0009C0
|                                *IPKAA501 03B228
|                                *IPKAA502 03B1F8
|                                *IPKAA503 03B1C0
|                                *IPKAA504 03B1E8
|                                *IPKAA505 03B1E8
|                                *IPKAA506 03B200
|                                *IPKAA507 03B1C8
|                                *IPKAA508 03B258
|                                *IPKAA509 03B258
|                                *IPKAA511 03B1E8
|                                *IPKAA512 03B210
|                                +IPKAA101 03AAD0
|                                +IPKAA102 03AB98
|                                +IPKAA103 03AC60
|                                 IPKAB000 03B378  03B378 001630
|                                +IPKAB100 03B388
|                                +IPKAB101 03B520
|                                +IPKAB103 03B58E
|                                +IPKAB102 03B5C0
|                                 IPKAG000 03B6B0  03B6B0 001638
|                                 IJJCPD2  03B6B8  03B6B0 001640
|                                *IJJCPD3  03B6B8
|                                 IPKAD000 03B8E8  03B8E8 001870
|                                *IPKAD101 03B8F8
|                                 IPKAD100 03B900  03B8E8 001888
|                                 IPKBA000 03BF80  03BF80 001F08
|ASSETA   03B8F0  03B8E8  03C388
|                                 IPKTA000 03B8E8  03B8E8 001870
|                                 IJ2M0093 03C010  03B8E8 001F98
|                                 IJJCPD1N 03C030  03B8E8 001FB8
|ASSECA   03BF88  03BF80  03F16F
|                                 IPKCA001 03BF80  03BF80 001F08
|                                +IPKCA998 03BF88
|etc.
└------------------------------------------------------------------------->
A + in front of the label indicates a referenced entry point;
an * shows an unreferenced label.
```

Figure 19. Example of a Linkage Editor Map Printout

INTERFACES

All librarian and supervisor services for retrieval of linkage editor library input and for storage of linkage editor library output are requested via CSECT INLPLEIT. This CSECT uses librarian and supervisor macros and interface control blocks. These are all listed in the operation description of INLPLEIT.

In the partition, buffers are provided for librarian services. They are discussed as "Buffers for Librarian Services" in the Chapter "Data Areas".

The stow table of the librarian is used by CSECT IJBCAT.

CROSS REFERENCES

Label-to-CSECT Cross Reference

The following labels are those which appear in the Sequence of Operation description in this manual.

The CSECT names are shortened where they begin by IJB to show only the last characters.

Label	CSECT
ABTERR	IJBLNK
ALNKOF	LNK
ALNKPR	LNK
CALCWA	INL
CANCL	LNK
CDENT1	LNK
CDSIZE	LNK
CESDENT	LNK
CNVHDX	LNK
CONCLIB	INL
CONNECT	INLPLEIT
CPHENT	IJBLNK
ELBCM	ESD
ELBER	ESD
ELBLD	ESD
ELBSD	ESD
ENDPRC	OTH
ENTCRD	SCN
ERROR	LNK
ESDRET	ESD
EXLOAD	LNK
FINDM	INLPLEIT
IJBLETR	IJBFIN
INCCRD	SCN
INLN000	INL
INTER100	INL
LCFIND	FIN
LCLOSE	FIN
LCSTBEG	LNK
LEITCALL	FIN
LEITPL	LNK
LNKNOT	FIN
LNKPOINT	FIN
LOGMSG	LNK
LTESID	LNK
LTMINE	LNK
NOTCTL	LNK
NOTEM	INLPLEIT
NXTCARD	IJBINL
OBJBUF	LNK
OPCATAL	INL
OPENC	INLPLEIT
OPENJ	INLPLEIT
PHAPBUF	IJBLNK
PHCRD	SCN
PHSPRC	CTL
POINTM	INLPLEIT
PRINT	IJBLNK
PRTLST	LNK
RDEXEX	LNK
RDNEXT	LNK
READ	INLPEIT
READIN	IJBFIN

Label	CSECT
REPPRC	OTH
RLDCON	RLD
RLDOR	RLD
RLDPRC	OTH
SCANCD	CAT
SHARBUF	LNK
SPACE1	LNK
SRCHCD	LNK
SRPCOD	LNK
STOWML	INLPLEIT
TESTCNT	IJBRLD
TSTUNR	RLD
TXTPRC	OTH
UPDATE	INLPLEIT
WRITE	INLPLEIT
WRLST	IJBRLD
WRTRFR	CTL
XTPHGT	LNK
XTPHNO	LNK

Message-to-CSECT Cross Reference

(only the causing CSECT is listed here)

Message	CSECT
2100	IJBOTH
01	SCN
02	LNK, SCN
10	SCN
11	SCN
12	SCN, INL
13	OTH, INL, SCN
14	SCN
15	FIN, MAP
16	SCN, LNK
17	SCN
20	CTL
21	SCN
22	CTL
23	CTL
24	CTL
25	SCN
32	SCN
33	SCN
35	INL
36	INL
39	ESD
40	ESD
41	ESD
42	ESD
43	ESD

Message	CSECT
44	LNK
45	ESD
46	ESD
47	OTH
50	LNK, OTH
51	OTH
55	OTH
56	OTH
58	OTH
60	CTL
2161	IJBSCN
70	OTH
80	INL
81	CTL
82	CTL
85	MAP
86	CAT
87	MAP
88	OTH
90	INL
91	OTH, INL
92	CAT
93	INLPLEIT
95	PLEIT
97	IJBLNK
99	MAP

Input-to-CSECT Cross Reference

Statement	CSECT
ACTION	IJBINL
INCLUDE	IJBSCN
PHASE	IJBSCN, IJBCTL
ENTRY	IJBSCN, IJBCTL

Input Record	CSECT
ESD	IJBESD
TXT	IJBOTH
RLD	IJBOTH
REP	IJBOTH
END	IJBOTH

Phase-to-Module Cross Reference

Phase	Module
$LNKEDT	IJBLE1 INLPLEIT

INDEX

READER'S
COMMENT
FORM

VSE/Advanced Functions
Diagnosis Reference: Linkage Editor
Order No. LY33-9112-0

This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Possible topics for comments are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name and mailing address:

____

____

____

What is your occupation? ______________________________

Number of latest Newsletter associated with this publication: ______________

Thank you for your cooperation. No postage stamp is necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page).

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

LY33-9112-0

Reader's Comment Form

Fold And Tape Please Do Not Staple Fold And Tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department 6R1
180 Kost Road
Mechanicsburg, PA 17055

Fold And Tape Please Do Not Staple Fold And Tape

IBM®

Cut or Fold Along Line

VSE/Advanced Functions Diagnosis Reference: Linkage Editor File No. S370/4300-31 Printed in U.S.A. LY33-9112-0

IBM

VSE/Advanced Functions Diagnosis Reference: Linkage Editor (File No. S370/4300-31) Printed in U.S.A. LY33-9112-0

LY33-9112-0